WEST

End of Result Set

Generate Collection Print

L38: Entry 1 of 1 File: USPT Nov 13, 2001

DOCUMENT-IDENTIFIER: US 6317748 B1
TITLE: Management information to object mapping and correlator

Brief Summary Text (5):
In general, network device agents provide such management information via a standard known as the Simple Network Management Protocol (SNMP). In accordance with SNMP, the management component uses Management Information Bases (MIBs) to describe the structure of the data available at the network device. MIBs are published structures of static information that relate to the dynamic information provided by a network device. For example, a particular MIB might describe information such as an association between a device's data entity and a name (object identifier), a definition of the data type for that entity, a textual description of the entity, how the entity is indexed, if the entity is a member of a complex data type and the access allowed for that entity. The management component then is able to describe the data made available on the SNMP device (e.g., in response to a query or event) by using an appropriate MIB module to model that data.

Detailed Description Text (27):
Note that since the FreeSpace parameter is highly volatile, in a typical implementation the value would be supplied by a provider 72, rather than being a stored or static value. As described in more detail below, the CIMOM 70 is capable of retrieving both the static and dynamic information from various sources including the CIM database 74 and/or appropriate providers 72, and returning the object instance to the application 58. In keeping with CIM, the database 74 and/or providers 72 provide their portion of the information without knowledge of its final destination, i.e., the identity of the client process, and the application 58 receives this information transparently, i.e., as instances of the class disk without knowledge of its actual source or sources.

Detailed Description Text (30):
As in relational database models, instance identification for a class capable of describing multiple instances (i.e., a keyed class) is done through the use of keys. Using the table and column model common to relation databases, the table corresponds to a CIM class, and the rows correspond to CIM instances. The column or columns which uniquely identify the row are designated as keys. Key properties are designated by attaching a qualifier named `key` to each property that constitutes the key for the class. Key qualifiers can be attached to more than one property to create a compound key. Using the MOF Syntax as an illustration, the Volume property has been designated as the key for the Disk class as shown in the table below:

Detailed Description Text (37):
A special type of property called a reference is supported within CIM. This property is a "pointer" to another class or instance within the schema. A reference takes the form of a string which contains an object path. A special type of class which contains a pair of references is called an Association. Instances of association classes are used to set up a binding relationship between two other objects. For example, while it is possible for each of objects A and B to contain reference fields which point to each other, this would entail designing the class to contain a reference property from the outset. In many cases, it may not be known at class design-time which objects may require references to each other. Instead, classes are ordinarily designed to contain no assumptions about relationships in which they may participate. Instead, an association class C can be created after classes A and B are created, wherein C can contain two reference property fields, one which is designated as pointing to instances of A, and the other as pointing to instances of B. In this manner, associations can be used to create object relationships without the knowledge or participation of the

objects themselves. Likewise, relationships can be removed without affecting the objects.

Detailed Description Text (38):
In a more specific example, it may sometimes be useful to associate a NetworkCard object with a NetworkProtocol object using a dedicated association class called NetCardToProtocolBinding. This is a desirable model, since a card may support more than one protocol, and the same protocol may be running over more than one network card on the host machine. The association objects describe the complex relationships without the more primitive objects being aware of it.

Detailed Description Text (59):
To create a range table 98, it is first noted that SNMP group object identifiers in the SMIR 86 are ordered lexicographically according to their OID, i.e., each component is essentially considered to be a single `character` having that component's numerical value. In other words, groups are sorted on a logical representation of the object identifier rather than on the string representation. For example, the group OIDs are ordered such that OID 1.1<1.2<1.2.1<1.2.1.5<1.3. From the MIB groups, a set of mutually exclusive ranges is created for each group (a group may have several other groups contained within its range). These ranges are ordered lexicographically, the result being the range table 98.

Current US Original Classification (1):
707/103X

Current US Cross Reference Classification (1):
707/103R

Current US Cross Reference Classification (2):
707/103Y

WEST

Generate Collection Print

L29: Entry 3 of 19 File: USPT Sep 18, 2001

DOCUMENT-IDENTIFIER: US 6292942 B1
TITLE: Manager for dynamic complex systems

Abstract Text (1):
A manager for a dynamic complex system is disclosed which associates a single, simple, versatile and malleable Management Scheme with each entity of the system, including the system itself. The Management Scheme permits management of systems via the mapping of information to a simple hierarchical management level wherein all of the information necessary to manage the system can be manipulated and communicated throughout the system via simple arithmetic operations. The Management Scheme is a compound integer number computed from management kernels and is stored in a Management Field of a Control Parameter. The Control Parameter also includes an Identifier Field storing a code number which identifies a particular entity. Management kernels are integer primes to an integer power of one or more.

Brief Summary Text (16):
At any given time, the characteristics of each entity (and, as noted, the system is considered an entity) are modified by parameters attributable to that entity. Parameters determine, at any given time, such things as the information content, state, or activities of an entity, but do not determine or affect the intrinsic nature of the entity. Parameters are, in general, variable.

Brief Summary Text (19):
Software, in general, is fuzzy, especially if not object oriented. Software can have unintentionally fuzzy characteristic, parameter, functionality and purpose sets due to various design inadequacies (bugs).

Brief Summary Text (20):
The Internet is an example of a catanet with a large software based constituency. The Internet can be considered a heterogeneous system because, from a management perspective, the membership functions of the characteristic and parameter sets of at least some entities can be fuzzy at any given time. The only practicable management system for the Internet would have to be highly specialized due to the complexity of the system itself and to the undetermined (or undeterminable) nature of the membership functions for elements of its constituent fuzzy sets of entity characteristics, parameters, functionalities and purposes.

Brief Summary Text (22):
U.S. Pat. No. 5,692,106 issued Nov. 25, 1997 to Simon Towers and Paul Mellor discloses a management method and apparatus for computer system tasks and services. An example of a service may be e-mail and tasks related to this services may be installing, configuring and diagnosing and removing faults. To facilitate the carrying out of a range of different types of management tasks in a computer system, declarative models are constructed of the various services provided by the system. These models specify the requirements that need to be met for the corresponding service to be available. These requirements are set out in terms of the system entities that need to be present and the inter-relationships of these entities. In addition, each management task is specified in a corresponding task program in terms of general inferencing operations that can be performed on any of the models. Execution of a particular management task involves carrying out inferencing operations on the appropriate service model in accordance with the task program for the management task under consideration.

Brief Summary Text (30):
In accordance with a further aspect of the present invention there is provided an electronic manager for an electrical, electronic, electro-mechanical or computer system comprising: a directive acceptance and result reporting layer for receiving an

indicator of each managed entity in the system and an associated control parameter, a manager instructions and rules of operation layer, a management kernel definition and redefinition layer which, together with the manager instruction and rules layer, has management kernels and associated management rules, each management kernel comprising a prime number factor raised to a positive integer power, a mathematical manipulation layer for dividing each control parameter by ones of the management kernels; a system entity information management layer responsive to the manager instructions and rules of operation layer, the management kernel definition and redefinition layer and the mathematical manipulation layer for deriving management rules; and a management plane communications capability layer responsive to the system entity information management layer for managing the system.

Drawing Description Text (1):
BRIEF DESCRIPTION OF THE DRAWINGS

Drawing Description Text (3):
FIG. 1 illustrates a Control Parameter in accordance with an embodiment of the invention.

Detailed Description Text (1):
DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Detailed Description Text (2):
The function of system management is to control the activities of system entities and of the system itself so that both entities and system can fulfill their respective purposes. The invention achieves this via a simple algorithm that maps hierarchical systems to what may be considered a flat (i.e., nonhierarchical) management plane. Entity and system activities are manipulated by the manager on the management plane via their characteristics and parameters and management instructions from a Management Intelligence Source (MIS). Once manipulation is complete, management directives flow back from the manager to the system to control and give direction to the system and entity activities. Due to the nature and simplicity of the algorithm, ongoing iterative corrections and adjustments can be made by the manager "on the fly". This permits most types of systems to be managed, including those that are fuzzy and extremely dynamic. The management of very complicated and extensive systems can be accommodated via management hierarchies.

Detailed Description Text (3):
With reference to FIG. 1, the manager requires that, regardless of tier, every entity of a system to be managed have a Control Parameter 102 comprising a unique Code Number C 108 in an Identifier Field 104 and a Management Scheme S 110 in a Management Field 106. It is not necessary that Code Number 108 be part of a single numbering scheme, only that it be unique. If Code Number 108 is not unique, then the manager requires that it be unique on the tier occupied and that the tier then be identified to the manager. Identifier Field 104 is a binary digit field long enough to accommodate any Code Number 108 used. Associated with each Identifier Field 104 is a Management Field 106. Management Field 106 is a binary digit field of sufficient length to accommodate Management Scheme 110. Management Scheme 110 embodies the essence of a plan or program of action to be followed by the manager and the system under differing management conditions. Management Scheme 110 is a positive integer number greater than zero that defaults to one if unassigned. Management Scheme 110 is comprised of a product of management kernels (k.sub.i.sup.n), S=.PI.k.sub.i.sup.n.

Detailed Description Text (4):
Management kernels (k.sub.i.sup.n) are the core of Management Scheme 110. Implicit in k.sub.i.sup.n is the information needed to inform the manager of characteristics and/or parameters and/or functions of each entity, as well as the information required by the manager to manipulate, associate and dissociate, reconfigure, communicate with and control entities on all tiers and to assign subsystem management hierarchies. To meet algorithm requirements, each k.sub.i must be drawn only from the set of integer prime numbers raised to positive integer powers n. This enables all k.sub.i.sup.n (except for k.sub.i.sup.0, which is "1") to be unequivocally and simply extracted from Management Scheme S 110 at any time or place and thus enables effective, versatile and dynamic management and management communications between the manager and the system together with all its entities.

Detailed Description Text (8):
A simplified view of the entities of system 200 is shown in FIG. 3. Each entity of the system, and the system itself, is given a Code Number, from G1 to G8, known to a

manager 300. Manager 300 may be a computer loaded with system management software for executing the method of this invention from software medium 310, which could be a disk, a tape, a chip or a random access memory containing a file downloaded from a remote source. From the Code Number C and associated Management Scheme S manager 300 can identify each entity and the system itself, together with entity or system characteristics, parameters and functions and can manage the system based on this information. Systems that are much more complex than this trivial example can be managed just as easily.

Detailed Description Text (10):
Manager 300 also receives from the Management Intelligence Source the table 402 (FIG. 4) dealing with the system 200 inputs 202, 204, 206 and the system output 210 as well as with the interpretation of the signals received via such inputs 202, 204, 206, and provided by the system 200 to such output 210. Thus, if a distinctive management kernel of numerical value the integer prime "5" is assigned to inputs 202, 204, 206, a prime number value for the management kernel 30 need not be assigned to the output 210. The numerical value for the management kernel assigned to output 210 can then default (for simplicity) to the numerical value "1". Similarly, if a distinctive management kernel of numerical value the integer prime "3" is assigned to the data inputs 202 and 204 a prime number value for the management kernel need not be assigned to the control input 206. The numerical value for the management kernel assigned to control input 206 can also default (for simplicity) to the numerical value "1". The parameter values in this simple example system 200 at the control input 206 must be differentiated in the FIG. 4 table 402 and are assigned the management kernels: "2" for a negative parameter value; "2.sup.2 " for a zero parameter value and "2.sup.3 " for a positive parameter value. The resultant numbers 5,3,1,2,4, and 8 are termed "management kernels" or "kernels" for the system.

Detailed Description Text (17):
Note that, where a choice of only two possible characteristics or parameters exists, a management kernel not equal to one need be assigned to only one of these characteristics or parameters, but not to both. In this example, there are only two kinds of inputs in the system, control inputs and data inputs. The management kernel "3" is assigned, by Management Intelligence Source 312 via table 402 (FIG. 4), to data and "1" is assigned to control because any input (indicated by management kernel "5") without a management kernel "3" (the data designator) assigned as well, must be a control input. The strategy of using a k.sub.i.sup.0 =1 designator for one choice from two possibilities is recommended in general to both reduce the number of operations required for system management and to reduce the length L of Management Field 106 (FIG. 1) to a minimum. Note also that, in general and for computational expediency, management kernels are best chosen from the set k.sub.i.sup.n, with n=1 (and with the additional use of k.sub.i.sup.0 =1): the choice of n>1 in this example is to illustrate alternative design choices that may suit certain implementations.

Detailed Description Text (18):
This method of identifying characteristics and parameters can be extended to dynamic systems whose inputs, or other entity characteristics or parameters, come and go rapidly and continuously. The method can also be applied to fuzzy systems where the closeness to presence or absence of a characteristic or parameter is indefinite to some extent as defined by the membership function of that characteristic or entity.

Detailed Description Text (19):
If a system to be managed has included an entity which in turn has an associated numeric parameter (N.sub.p) requiring management, the numeric parameter may be included as a factor in the management scheme for the system. If more than one entity has an associated numeric parameter, the numeric parameters of each such entity can be included in a subsystem via assignment of a unique hierarchy of subsystem identifiers, k.sub.u1.sup.n, k.sub.u2.sup.n, k.sub.u3.sup.n, . . . , to the management scheme for these entities.

Detailed Description Text (20):
In either case, if the largest possible value of "S" in the system ("S.sub.max ") and the largest possible value of N.sub.p in the system ("N.sub.p max ") are such that S.sub.max.times. (N.sub.p max).sup.2 <about 1.times.10.sup.100, each numeric parameter value, N.sub.p, required can then be stored in Management Scheme S of the system or subsystem, in addition to any required k.sub.i.sup.n, using the following protocol: If N.sub.p values must be included in S as a factor, then N.sub.p must first be squared so as to render it unambiguously extractable by the manager,
S=(.PI.k.sub.i.sup.n).times.N.sub.p.sup.2 with n now restricted to either "0" or "1".

For example, if the kernels for a particular entity in a system or sub-system are 2, 3 and 5 (with 7 an unused but valid kernel), if values of n other than "0" or "1" are permitted, and if N.sub.p =14 is to be included in S as is, i.e., S=.PI.k.sub.i.sup.n.times.N.sub.p. S becomes, in this example: S=(2.times.3.times.5).times.14=420. Extracting management kernels, the manager computes: 420=2.times.2.times.3.times.5.times.7 and incorrectly concludes that: k.sub.i.sup.n =(1,)2,3,5,7 resulting in the loss of the numerical value of N.sub.p. But if, in the same example, n is restricted to either "0" or "1" and N.sub.p is entered in S as: N.sub.p.sup.2 =14.sup.2 =196, S becomes: S=(2.times.3.times.5).times.196=5,880. Extracting management kernels, the manager computes:

Detailed Description Text (21):
Taking the square root of N.sub.p.sup.2 =196, the manager correctly interprets the management kernels as 2, 3, 5 and the numeric parameter value as 196=14.

Detailed Description Text (22):
The use of subsystems to store numeric parameters removes the restriction on the value of n for k.sub.i.sup.n which would otherwise be imposed. By confining numeric parameters to numeric subsystems, n may then be a positive integer greater than or equal to zero outside of any subsystem defined to store numeric parameters. In numeric subsystems themselves, at the bottom of the hierarchy of subsystem sets (i.e. the jth set, wherein the N.sub.p values themselves are stored), the n in k.sub.i.sup.n must be a positive integer equal to zero or one only. Very complicated systems may be subdivided into any type (or types) of subsystems to simplify management as well as to store numeric parameters.

Detailed Description Text (35):
Layer 1004 deals with information access, storage and retrieval as it pertains to entities in system 1012. With reference to FIG. 1, for each entity to be managed, information is stored in a Control Parameter 102 as an Identifier Field 104 containing the Code Number C 108 together with an associated Management Field 106 containing the Management Scheme S 110. In the simplified exemplary system this layer represents a complete set of characteristics and parameters for inputs, outputs and operations. FIG. 5 illustrates a Code Number and Management Scheme associated with each entity in the exemplary system.

Detailed Description Text (39):
Layer 1002 is a decoding and encoding layer. At layer 1002, mathematical operations take place to extract to encode the management kernels, k.sub.i.sup.n, and any numeric parameters N.sub.p from the management scheme assigned to each code number.

Detailed Description Text (42):
Systems may be subdivided, in reality or in thought only, into subsystems to enable management of very complicated and/or extensive systems or to accommodate systems where numeric parameter values are assigned to entities included in the system. Referring to the telecommunications example of FIG. 11, a manager 1102 in a complicated system, such as a backbone network 1104, can manage more effectively and more efficiently if the member collector networks 1106, 1108, included local networks 1110, 1112, 1114, 1116 and nodes 1120, 1122, 1124, 1126, 1128, 1130 are each treated as subsystems of the backbone network system 1100. As is known from, or is implicit in, Class Theory, hierarchical relationships are essential amongst the system entities to be managed or management is impossible.

Detailed Description Text (54):
Parameter Values cannot Violate Characteristics.

Detailed Description Text (55):
Independent of the nature of the system being managed, this rule is embedded in the Manager Instructions and Rules of Operation at layer 1006 and cannot be violated. Characteristics are distinctive activities or actions that reflect the inherent nature of each entity. The characteristics of each entity are modified by parameters attributable to that entity. Parameters determine, at any given time, such things as the information content, state, or activities of an entity. Characteristics cannot be perverted by the form, configuration or the function of an entity else the entity will not work as per design intent.

Detailed Description Text (56):
A characteristic of an entity may be a one litre capacity. A parameter corresponding to this characteristic may be the volume currently held in the entity. To restate General

Rule #4 in relation to this example, one cannot hold two litres of water in a one litre container.

Detailed Description Text (61):
Class or Set Parameters cannot Violate the Parameters of their Members or Included Sets or Entities.

Detailed Description Text (64):
For subsystems which do not have numeric parameters, while kernels may be represented by k.sub.i.sup.n (with n being a positive integer), n may ideally be set at "1" for all kernels making the set of k.sub.i.sup.n more homogeneous and (arguably) easier to process and interpret.

CLAIMS:

6. The method of claim 5 wherein, for each said kernel, said positive integer power has a value of one and said management scheme further comprises a squared numeric parameter as a factor in said product of said management kernels.

8. The method of claim 7 wherein, for each kernel included in said management scheme having said unique subsystem identifier kernel, said positive integer power has a value of one and wherein said management scheme further comprises a squared numeric parameter as a factor in said product of said management kernels.

13. An electronic manager for an electrical, electronic, electro-mechanical or computer system comprising:

a directive acceptance and result reporting layer for receiving an indicator of each managed entity in said system and an associated control parameter;

a manager instructions and rules of operation layer;

a management kernel definition and redefinition layer which, together with said manager instruction and rules layer, has management kernels and associated management rules, each management kernel comprising a prime number factor raised to a positive integer power;

a mathematical manipulation layer for dividing each control parameter by ones of said management kernels;

a system entity information management layer responsive to said manager instructions and rules of operation layer, said management kernel definition and redefinition layer and said mathematical manipulation layer for deriving management rules; and

a management plane communications capability layer responsive to said system entity information management layer for managing said system.

WEST

End of Result Set

Generate Collection Print

L69: Entry 13 of 13 File: USPT Jul 29, 1997

DOCUMENT-IDENTIFIER: US 5652880 A
**** See image for Certificate of Correction ****
TITLE: Apparatus and method for storing, retrieving and presenting objects with rich links

Abstract Text (1):
A method and apparatus for managing and sub-classifying units of data (objects) through the use of codified linkages between objects and for querying and presenting such objects, their linkages, and the codified data concerning their linkages. The query and presentation method is called Query-By-Outline (QBO). The method of retrieving and displaying interrelated data stored according to class memberships, linkages between individual objects, or linkages between classes, upon identification of an object of interest, including the steps of (1) creating an inventory of related objects related to the object of interest, the inventory including information about the objects themselves as well as relationship information about their relationships to the object of interest; (2) developing an arrangement of the inventory making use of the relationship information as well as information about the related objects themselves; (3) simplifying the arrangement for presentation by intelligently grouping the inventory into groups determined at least in part by the relationship information and the information about the related objects; and (4) displaying the arrangement.

Brief Summary Text (2):
This invention relates generally to linking and querying linked data in a computer system, and more specifically to managing and sub-classifying units of data, called objects, through the use of codified linkages between objects, and a system for querying and presenting such objects, their linkages, and the linkage data so codified.

Brief Summary Text (4):
Although the prior art provides methods for sub-classifying and cross-referencing data, as well as for querying such sub-classifications and cross-references, it suffers from a number of major shortcomings. One major shortcoming of prior art systems for cross-referencing data is that they do not provide for the codification of links to allow each link classification to store a codified set of user-defined fields relevant to that classification itself. Prior art systems exist for linking multiple units of data. For example, data on Suzanne and data on her husband Jerry may be linked.

Brief Summary Text (5):
Prior art systems allow, for example, a link to be established or programmed between a specific person and a specific school. The link itself contains only the type of link ("Pupil" in this case) and some internal information such as record or object numbers, offsets or record keys to maintain the link between the subject records of the link. Two such "Pupil" links are depicted in FIG. 3.

Brief Summary Text (6):
Prior art systems suffer from a number of limitations. First, they do not provide for user-specified codification of the links themselves. They provide only for the codification of the records in the tables. This forces the user to do considerable custom programming to facilitate the storage of such link information, or suffer increased difficulties accessing and querying such link information if the information is stored free-form and not codified.

Brief Summary Text (7):
Such custom programming is an extensive task when multiple links of various types and

codifications are to be provided from any number of individual records in any number of different tables, using any number of different file types and fragments in any expanse of computer store, especially when the links are many and/or contain considerable quantities of information. Prior art systems also require substantial on-going re-programming and maintenance to accommodate information pertinent to links as new link types are needed and as the nature of link information changes.

Brief Summary Text (9):
The method and apparatus of the present invention allows links to be classed and codified based on their classifications, so that, in the above example of married couple Suzanne and Jerry, the link between them may contain fields of information such as wedding date, wedding place or divorce date, since the link was of the class Husband/Wife. The system of the present invention can also enable a link, such as of the classification Husband/Wife, to automatically be a member of a parent link class such as "Family" that is further codified to include fields of information pertinent to all "Family" type links.

Brief Summary Text (10):
Briefly, the method of the invention of retrieving and displaying interrelated data stored according to class memberships, linkages between individual objects, or linkages between classes, upon identification of an object of interest, includes the steps of: (1) creating an inventory of related objects related to the object of interest, the inventory including information about the objects themselves as well as relationship information about their relationships to the object of interest; (2) developing an arrangement of the inventory making use of the relationship information as well as information about the related objects themselves; (3) simplifying the arrangement for presentation by intelligently grouping the inventory into groups determined at least in part by the relationship information and the information about the related objects; and (4) displaying the arrangement.

Brief Summary Text (11):
As prior art systems do, this invention allows each link to be classified, such as "Pupil" and "Instructor". Where this invention advances the art is that it provides a significantly easier to implement and more concise way to express link information and sub-class data by allowing links to be codified and therefore to be able to store data relevant to the links themselves. The links themselves are treated as codified records with corresponding data dictionary entries or data definitions that define their codification by link type. Thus, each link type has a separate table with each record in the table storing information about one link of that link type. This arrangement will be called "rich" link information.

Brief Summary Text (12):
Modeling and managing such rich link information for more than a few objects and links is not possible with prior art systems without costly and time-consuming custom programming and/or the ineffective query methods required by the non-codified free-form storage of link information.

Brief Summary Text (13):
Prior art flat-file and relational database users are required: (1) to employ programmers to program additional tables for the link codification data; (2) to provide program logic to assign unique arbitrary identifications to every link and every object referenced by each link; and (3) to provide custom programming each time link types or their codifications are added or changed. Prior art object-oriented databases require the programming of links through similar means and through the use of the sub-classing capabilities of such databases. This invention also provides a superior method of sub-classing data where such sub-classification is, or can be expressed as, the result of linkages between objects. For example, a person named Nancy might have the properties of a person, but if she is linked to a company named ACME Sciences with a link of the link type employee, and the link is codified to contain information about her employment there, then as a result of this rich link she has been effectively sub-classed to be an employee, with all the attendant properties of being an employee stored on the employee link. Likewise, for each other link Nancy has she will be sub-classed by each link's link type and will inherit the properties codified for each such link.

Brief Summary Text (14):
At best, prior art methods only provide tools for programming the storage and "a priori" presentation of such information on a case-by-case basis, the ability for a user to outline such information, traditional filter-and-display techniques and

graphical web views of links that become impractical as the number of links grows.

Drawing Description Text (4):
FIG. 3 illustrates the prior art linking of records between two tables.,

Drawing Description Text (5):
FIG. 4 illustrates the codification of the invention of "Pupil" links between records in two tables.

Drawing Description Text (6):
FIG. 5 illustrates the codification of the invention of "Instructor" and "Husband/Wife" links.

Detailed Description Text (16):
Linkage class: Any class whose members are links.

Detailed Description Text (20):
An object parent class of "Place" might include object sub-classes such as "Hotel", "Church" and "Restaurant". While "Restaurant" is an object sub-class of "Place", it also may be an object parent class of the object sub-classes "Pizza parlor" and "Care". A particular object, such as "Hard Rock Cafe at 1234 Main Street" might be a member of the object sub-class "Cafe", making it automatically also a member of the object parent class "Restaurant" and, in turn, the object parent class of "Restaurant"--"Place".

Detailed Description Text (21):
Continuing the example, another object such as "Jill Jones" might be a member of the object sub-class "U.S. Citizen" and so automatically also be a member of the object parent class of "U.S. Citizen"--"Person". Jill may be linked to the Hard Rock Cafe as its "Manager". She therefore would be a member of the linkage sub-class "Manager" and automatically also a member of the "Manager's" linkage parent class "White collar worker", and also the linkage parent class of "White collar worker"--"Employee". The linkage of "Manager" might be codified to include years of management experience and the date of last management certification, and so forth. The linkage of "White collar worker" might be codified to include information about the company-provided vehicle, while the linkage of "Employee" might contain such information as the date hired, date terminated, reason for leaving, graphic image of the employee's employment application, and so forth.

Detailed Description Text (24):
The following detailed description of the method of the invention is divided into several sections. The first section discloses the system for implementing codified links. Later sections describe the techniques for querying and displaying these links along with their codified information.

Detailed Description Text (27):
Management of Rich Links

Detailed Description Text (28):
Many computer programs allow for the linking of information. Relational databases allow records in one table to link to records in another table. Hypertext programs allow words, phrases, or graphics to be linked to other words, phrases, or graphics. Word processing programs allow footnotes to be linked to body text. Object oriented databases allow objects and classes to be linked to other objects or classes. All of these technologies provide links that contain the information needed to classify and maintain the linkages.

Detailed Description Text (29):
The system of the present invention differs in that when two or more objects are linked, the link is not only classified but it is codified to include user-defined fields of information pertinent to that classification. This codification of links allows them to represent all of the information concerning the relationships represented by each classification of link and effectively to provide for the sub-classification of the objects that are the respective subjects of the links, which is a new and unique method for sub-classing objects.

Detailed Description Text (30):
For example, under some prior art systems there may, for example, be a record in a "Persons" table (representing a particular person) and a record in a "Schools" table (representing a particular school) and a desire to link a specific person in the

"Persons" table to a specific school in the "Schools" table. The "Persons" and "Schools" tables are typically codified to include various user-defined fields to be tracked for each record in those tables. An example of the information used in the codification of such "Persons" and "Schools" tables is depicted in FIG. 2.

Detailed Description Text (31):
FIG. 4 illustrates how this invention codifies links to include relevant information about each link. New link types and their codifications can be added and changed interactively by non-technical computer operators, machines, or software allowing the expression of an evolving network of rich link relationships without the need for special computer programming. In FIG. 4 the "Pupil" type link between the Person record for Janis Smith in the Persons table and the School record for Coronado Grade school in the Schools table is codified to include her enrollment date, graduation date and grade point average. Her Pupil type link to Sunnyslope high school is similarly codified. These two Pupil type links and all other Pupil type links are stored in a link record table for the link type of pupil. Each link record in the Pupil link table is codified to include the information pertinent to Pupil links in the same manner as each record in the Persons table is codified to include information pertinent to each Person record. Thus, just as prior art codified tables accommodate the storage of codified fields of information for object records, this invention's link record tables accommodate the storage of codified fields of information for the link records.

Detailed Description Text (32):
FIG. 5 depicts how this invention facilitates codified links between any objects regardless of type. An object may have none, one, or many links to another object or to many other objects. These form a network, or web, of links from any object to any other object. Codified links also facilitate the expression of other information about relationships, including temporal information, such as the time of their commencement and expiration. In FIG. 5 the short marriage between John, an instructor at Sunnyslope, and Janis, a pupil there, is shown in the Husband/Wife link table. They were married at Grace Cathedral on Oct. 1, 1990, four months after Janis graduated (as shown in FIG. 4 in the Pupil link table), but divorced in six months.

Detailed Description Text (33):
FIG. 5 also illustrates that John Smith is a member of the "Person" object class since his record is in the "Persons" table. He is linked to Sunnyslope High School by a link that is of the class "Instructor". This link enrolls him as a member of the linkage class "Instructor", which gives him the additional properties of "Date of hire", "Termination date" and "Direct dial phone", which are codified fields associated with and stored on that link. Notice that John, a person, has been effectively sub-classed as an instructor in two instances, and a husband in one instance by his respective links to the two schools Sunnyslope and Harvard, and to his wife Janis, while each link stores the appropriate additional properties for each sub-classification of John.

Detailed Description Text (34):
Multiple instances of like link types and their corresponding sub-class memberships are automatically handled by this invention since each link exists separately and stores the data relevant to that link. Therefore, since John teaches at two schools, Sunnyslope and Harvard, each of these two Instructor links contains his date of hire, termination date and direct dial phone number relevant to John's employment at that particular school.

Detailed Description Text (35):
These codified links are superior to the links provided in prior art hypertext systems since their codification facilitates the storage of user-defined fields of information pertinent to each link. They are also superior in many situations to the object sub-classing schemes employed by prior art object-oriented databases where such sub-classification is, or can be expressed as, the result of linkages between objects. The codified links of this invention are able to more easily and more concisely map to the human understanding and expression of interrelated information and can be established and maintained by non-technical computer operators, machines, or software without custom computer programming.

Detailed Description Text (37):
The following sections disclose the feature of the invention that makes use of the links, codified links, object sub-classing and link sub-classing of the invention to provide a query method called Query-By-Outline (QBO). These sections will reference the flow charts provided in FIGS. 14 through 22.

Detailed Description Text (40):
Once the object of interest has been designated, a test is made in step 102 to determine if there are any links connected to the object of interest. If there are none, step 106 is the next step taken after step 102 and the Output Temporary Storage Buffer (O-TSB) 16 is displayed. If step 102 instead determines that there are any links connected to the object of interest, then step 103 is the next step after step 102.

Detailed Description Text (48):
So, if there are any links connected to the object of interest, step 102 directs program control to step 103, which calls the routine named "Construct Data in I-TSB", causing step 120 shown in FIG. 15 to be executed. Step 120 is the entry point of the "Construct Data in I-TSB" routine of FIG. 15 and directs control to step 121.

Detailed Description Text (54):
b. Referring to FIG. 16, step 141 positions to the first link connected to the object of interest.

Detailed Description Text (55):
c. Step 142 retrieves the link information of the link connected to the object of interest.

Detailed Description Text (56):
d. Step 143 retrieves the information about the object referenced by the link.

Detailed Description Text (57):
e. Step 144 appends I-TSB 14 with a new entry that is a copy of the retrieved link and object information.

Detailed Description Text (58):
f. Step 145 determines if there is another link connected to the object of interest.

Detailed Description Text (59):
g. If step 145 determines that there is another link connected to the object of interest, then program control is looped back to step 142 and that link's link information is retrieved and the process is repeated for each link connected to the object of interest.

Detailed Description Text (60):
h. When step 145 determines that there are no more links connected to the object of interest, then program control is directed to step 146, the "Return" step of the "Collect Inventory" routine, causing program control to be directed to the next step after the step that called the routine, which was, in this example, step 123 in FIG. 15.

Detailed Description Text (62):
a. Create groups of objects having like linkage sub-classes to the object of interest for each like linkage sub-class. Two or more objects must qualify in order to create these groups. For example, if two objects were each linked as "Daughter" to the object of interest, then these two objects are grouped together into one group for the linkage sub-class of "Daughter".

Detailed Description Text (63):
b. Create groups of objects having like linkage parent classes to the object of interest for each like linkage parent class. Two or more objects must qualify in order to create these groups. For example, if one object is linked as "Sister" to the object of interest and another object is linked as "Brother", and, furthermore, "Sister" and "Brother" were linkage sub-classes belonging to the linkage parent class "Family", then these two objects are grouped together into one group based on the linkage parent class of "Family".

Detailed Description Text (64):
c. For the remaining objects linked to the object of interest that have not been grouped, create groups of objects belonging to like object sub-classes for each like object sub-class. Two or more objects must qualify in order to create these groups. For example, if the object of interest are linked to two objects that are both members of the object sub-class "Hotel", then these two objects will be grouped together into one group.

Detailed Description Text (65):

d. Create groups of objects belonging to like object parent classes for each like object parent class. Two or more objects must qualify in order to create these groups. For example, if the object of interest is linked to one object that is a member of the object sub-class "Church" and to one object that is a member of the object sub-class "Health club", and where "Church" and "Health club" are sub-classes belonging to the object parent class "Place", then these two objects will be grouped together into one group based on the parent class "Place".

Detailed Description Text (66):
e. The remaining objects linked to the object of interest that have not been grouped by the above grouping methods remain as individual objects in I-TSB 14.

Detailed Description Text (68):
Step 161 sorts the entries in I-TSB 14 by linkage sub-class.

Detailed Description Text (69):
Step 162 sub-sorts (that is, sorts the entries without disturbing any previous sort or sub-sort of the entries) the entries in I-TSB 14 by linkage parent class.

Detailed Description Text (76):
Step 168 uses the examination information from step 167 and determines whether the entry is of the same linkage or object class or sub-class as the last examined entry-that is, is the entry of the same linkage class, of the same object class, of the same linkage sub-class or of the same object sub-class as the last examined entry? If the entry is of the same linkage or object class or sub-class as the last examined entry, then program control is looped back to step 166 and the process is repeated until all entries in I-TSB 14 have been processed. If the entry is not of the same linkage or object class or sub-class as the last examined entry, then program control proceeds to step 169.

Detailed Description Text (78):
Step 170 inserts an "End-of-group" marker entry in I-TSB 14 so that entries of like linkage or object classes or sub-classes are grouped together by such group markers in the. I-TSB 14. Then program control is looped back to step 166 and the process is repeated until all entries in I-TSB 14 have been processed.

Detailed Description Text (90):
a. Step 241: Sort the headings in H-TSB 15 by the most senior classes of the objects represented by each heading. For example, in a collection of data organized with the three most senior parent classes of "Person", "Place", and "Thing", the headings whose objects belong to the "Person" class or any sub-class of "Person" are arranged together in H-TSB 15; headings whose objects belong to the "Place" class or any sub-class of "Place" are arranged together in H-TSB 15; the remaining "Thing" headings are arranged together in H-TSB 15.

Detailed Description Text (91):
b. Step 242: Within each of the most senior classes (Persons, Places, and Things in this example), place the headings in the following order:

Detailed Description Text (92):
(1) Step 244: Headings that represent objects having like linkage sub-classes to the object of interest (e.g., "Sister", where "Sister" is a linkage sub-class belonging to a linkage parent class such as "Family");

Detailed Description Text (93):
(2) Step 245: Headings that represent objects having like linkage parent classes to the object of interest (e.g., "Family");

Detailed Description Text (94):
(3) Step 246: Headings that represent objects belonging to like object sub-classes (e.g., "Hotel", where "Hotel" is an object sub-class belonging to an object parent class such as "Place");

Detailed Description Text (95):
(4) Step 247: Headings that represent objects belonging to like object parent classes (e.g., "Places").

Detailed Description Text (106):
For example, if an entry representing one or more objects belonging to the object

sub-class of "Hotels" already exists in O-TSB 16, and another entry existing in O-TSB 16 represents one or more objects belonging to the object parent class of "Places" (the parent class of "Hotels"), then the entry "Places" is transformed to "Other places". This transformation is done by rules incorporated in the software or learned by or programmed into the software and/or were taught or overridden by humans, machines or other software during the course of operation of the software, and the entry "Other places" is positioned below or subordinate to the entry "Hotels".

Detailed Description Text (107):
8. Steps 276 and 277 transform to the past tense each entry in the O-TSB 16 that represents one or more objects whose linkage to the object of interest has expired based on the temporal information about the linkage, or whose linkage represents a discrete event that has occurred at some prior time (for non-textual or non-English language implementations, modify the heading with the appropriate symbols, graphics or sounds to cause the same effect). For example, an entry of "Works at" would be transformed to "Worked at". This transformation is done by rules incorporated into the software or learned by the software by the operation of the software and/or were taught or overridden by humans, machines or other software during the course of operation of the software.

Detailed Description Text (108):
9. Steps 278-279 transform to future tense each entry in O-TSB 16 that represents one or more objects whose linkage to the object of interest has yet to commence based upon the temporal information about the linkage, or whose linkage represents a discrete event that is to occur at some future point in time (for non-textual or non-English language implementations, modify the heading with the appropriate symbols, graphics or sounds to cause the same effect). For example, an entry of "Works at" is transformed to "Will work at". This transformation is done by rules incorporated into the software or that were learned by the software by way of operation of the software and/or taught or overridden by humans, machines or other software during the course of operation of the software.

Detailed Description Text (109):
10. Step 280 modifies each entry in O-TSB 16 with a cue, symbol, graphic, or other delineation that indicates whether that entry (1) represents a collection of objects, (2) represents an object that is linked to another object or objects, (3) represents an object that is an atomic entry (one that has either no links or has only one link and that link is to an object already displayed in the output as its immediate parent), (4) that does not represent a collection of objects and is not linked to any other object or objects, (5) represents an object outside the domain or environment of the system in which this invention is implemented, (6) represents an action or executable software procedure that will occur if the entry is activated, (7) represents a quantity of information about the entry or its links, or (8) represents other information about the entry, such modification providing a cue about such information to the human, machine or software.

Detailed Description Text (111):
Display Step 106 of FIG. 14 represents the contents of O-TSB 16 by displaying, printing, transferring, communicating or otherwise outputting them for a user, machine, or software to use. The user may request and receive additional information about any of the entries in O-TSB 16, including any objects that such entry represents, any linkages that such entry represents or any other information stored or otherwise available about such entry, objects or linkages. The user, machine, or software can also create, change, delete, or manipulate information about any of the entries in O-TSB 16, including any objects such entry represents, any linkages such entry represents or any other information stored or otherwise available about such entry, objects or linkages. The additional information may be delineated through the use of graphics, indenting, symbols, sounds or otherwise. Step 107 determines whether the user, machine or software wishes to identify another object of interest, causing the entire process heretofore described to begin again on the newly identified object of interest.

Detailed Description Text (112):
The system of the present invention analyzes an arbitrarily complex network of codified links and reduces it to an outline so that the user is presented with a shorter list of more general path selections from which to query, bifurcate, explore or manipulate.

Detailed Description Text (115):
Referring to FIG. 8, even though John Doe is linked to fifteen other objects, the QBO

technique of this invention organizes and stages the links, using the typical statistical distribution of the links, into their various classes, distilling them into fewer, more easily digested subgroups and applies temporal, pluralization and other modifications to the subgroup headings to best communicate the summarization and content to the user.

Detailed Description Text (116):
FIG. 8 depicts the output using the outline of FIG. 7 where the user requested that all of the subheadings presented under John Doe in FIG. 7 were to be expanded to reveal the objects and sub-headings each heading represents. Notice in FIG. 8 that some objects, such as John's memberships in the Association for Computing Machinery and MENSA are grouped by their link class of membership, while others, such as his vehicles, are grouped by their object classification of vehicles, while others, such as his mother and sisters, and his horse and dog are grouped by their broader parent link and object classes of family and animals, respectively.

Detailed Description Text (117):
The link class of "Busboy", that ties John Doe to the Hard Rock Cafe, has been transformed to the past tense by the method of this invention by the addition of the word "Former", as shown in FIG. 8. The method of this invention automatically transforms to past or future tense, as required, the links that have expired, have yet to commence, or are scheduled to commence again based on user-defined temporal information stored in the codified links.

Detailed Description Text (119):
Notice in FIG. 9 how the method of the invention automatically replicates or clones the objects Cadillac Allante as a product of Cadillac Corporation and John Doe as a Cadillac Allante owner to show them from all relevant perspectives simuitaneously. This is the result of their being inventoried and outputted each time they are referenced by some object of interest, even when they may already appear in the output.

Detailed Description Text (120):
The entire perspective of the output may be changed at the user's request. For example, the user may designate General Motors as the new object of interest, and request that General Motors be hoisted to the focal-point of the output. This process of hoisting General Motors causes all information linked to General Motors, whether presented at all, presented above or presented below it in the prior output, to be organized subordinate to and from the perspective of General Motors. FIG. 10 illustrates the output resulting from hoisting General Motors and then expanding various subordinate entries to the object of interest, General Motors, to reveal the information represented by those subordinate entries.

Detailed Description Text (121):
FIG. 11 illustrates the output after hoisting Cadillac Allante into the primary focus of the output. Hoisting causes the output to be cleared and Cadillac Allante to be inserted at the top of the output. The links from Cadillac Allante are inventoried, as are the objects referenced by the links, and the subordinate entries of "Owners" and "Manufacturer, Cadillac Corporation" are created. The "Owners" entry represents a group of objects, John and Richard Doe, both of whom are linked to Cadillac Allante with links of the class "Owner" while the "Manufacturer, Cadillac Corporation" entry represents the link type of "Manufacturer" and the object of the "Manufacturer" link, "Cadillac Corporation". The links from Cadillac Corporation are inventoried and summarized under Cadillac Corporation as the subordinate entries "V. P. Marketing, Tanya Murray", "Headquarters, General Motors" and "Other products". The entries "Headquarters, General Motors" and "Other products" have been similarly expanded to reveal the objects or groups they represent.

Detailed Description Text (122):
FIG. 12 depicts the output when Richard Doe is hoisted into the focus of the output. The entry "Documents" represents a group of linked objects having the same object parent class of "Document". Some of the objects are of the object sub-class "Letter" and some of the object sub-class "Personal". The "Personal" heading has been expanded to reveal the two personal documents linked to Richard Doe. The additional information in the output for the first personal document indicates that the object consists of a file named RESUME.DOC last updated on Oct. 28, 1989, while the second personal document is a file fragment 97,316 characters in length last updated on Nov. 15, 1989.

Detailed Description Text (133):
QBO implements basic outlining functions, such as expanding and collapsing subordinate

levels, but creates the outline for the user from the network of connections that are defined. Not only is the user not required to construct fixed outlines of his/her data from the beginning, but QBO implements a hoist function that hoists a selected object to the focus of the outline and then automatically outlines all of its connections with subordinate headings causing a re-hierarching of the network of information from the perspective of the hoisted object. The hoist facilities in prior art outlining systems hoist an object up to the top, but do not re-outline everything, including all objects linked to the hoisted object, whether such objects are presented at all, presented above or presented below the object. The hoist facilities in prior art outlining systems operate more like pruning facilities that remove any information related to the object which is not displayed below the object instead of providing a refocus and presentation of all related information, as is provided by this invention.

Detailed Description Text (135):
The QBO feature of the invention is more powerful, more automatic and provides new capabilities that capitalize on the capability of the method of this invention to support and interpret arbitrary networks of classified, sub-classified and codified links between objects.

Current US Original Classification (1):
707/103R

CLAIMS:

1. A computer implemented method of retrieving and displaying interrelated data stored according to class memberships, linkages between individual objects, or linkages between classes, upon identification of an object of interest, comprising:

retrieving information related to objects related to said object of interest wherein at least one related object is related to said object of interest by a rich link data structure;

accessing the rich link data structure to retrieve rich link relationship information including information associated with both the object of interest and the related object;

developing an arrangement of said related objects making use of the relationship information as well as information about said related objects themselves;

simplifying said arrangement for presentation by grouping said related objects into groups subordinate to said object of interest and determined at least in part by said relationship information and said information about said related objects; and

displaying said simplified arrangement.

2. A computer implemented method of retrieving and displaying interrelated data stored according to class memberships, linkages between individual objects, or linkages between classes, upon identification of an object of interest, comprising:

retrieving information related to objects related to said object of interest wherein at least one related object is related to said object of interest by a rich link data structure;

accessing the rich link data structure to retrieve relationship information including information associated with both the object of interest and the related object;

developing an arrangement of said related objects making use of the relationship information retrieved from the rich link data structure as well as information about said related objects themselves;

simplifying said arrangement for presentation by grouping said related objects into groups subordinate to said object of interest and determined at least in part by said relationship information and said information about said related objects; and

displaying an outline of said simplified arrangement.

3. A computer implemented method of retrieving and displaying interrelated data stored according to class memberships, linkages between individual objects, or linkages between classes, upon identification of an object of interest, comprising:

retrieving information related to objects related to said object of interest wherein at least one related object is related to said object of interest by a rich link data structure;

accessing the rich link data structure go retrieve relationship information including information associated with both the object of interest and the related object;

developing an arrangement of said related objects making use of the relationship information retrieved from the rich link data structure as well as information about said related objects themselves;

simplifying said arrangements for presentation by grouping said related objects into groups subordinate to said object of interest and determined at least in part by said relationship information and said information about said related objects;

displaying an outline of said simplified arrangement; and

responding to an identification of a second object of interest by redisplaying said outline of said simplified arrangement to show the relationship between said second object of interest and the other related objects which may include the original object of interest, wherein the other related objects which may include the original object of interest are subordinate to said second object of interest.

4. A computer implemented method of retrieving and displaying interrelated data store according to class memberships, linkages between individual objects, or linkages between classes, upon identification of an object of interest, comprising:

retrieving information related to objects related to said object of interest wherein at least one related object is related to said object of interest by a rich link data structure;

accessing the rich link data structure to retrieve relationship information including information associated with both the object of interest and the related object;

developing an arrangement of said related objects making use of the relationship information retrieved from the rich link data structure as well as information about said related objects themselves;

simplifying said arrangement for presentation by grouping said related objects into groups subordinate to said object of interest and determined at least in part by said relationship information and said information about said related objects; and

displaying an outline of said simplified arrangement, the outline using syntax to reflect at least one aspect of the nature of the relationship between the objects in said simplified arrangement.

8. A computer implemented method of storing and subsequently displaying interrelated data stored according to class memberships, linkages between individual objects, or linkages between classes, comprising:

retrieving information related to objects related to said object; of interest wherein at least one related object is related to said object of interest by a rich link data structure;

accessing the rich link data structure to retrieve relationship information including information associated with both the object of interest and the related object;

developing an arrangement of said related objects making use of the relationship information retrieved from the rich link data structure as well as information about said related objects themselves;

classifying said relationship information into relationship classifications, each of which is subordinate to said object of interest;

attaching to each of the relationship classifications data pertinent to that classification of said relationship;

displaying said interrelated data including said relationship classifications and their

attached data; and

displaying said arrangement.

WEST Search History

DATE: Thursday, August 21, 2003

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L65	L64 and link$	13	L65
L64	L63 and place and time	20	L64
L63	l32 and (wedding)	24	L63
L62	l32 and (wedding near video)	0	L62
L61	L60 and ((build$ or generat$) near graph)	4	L61
L60	L59 and (compar$ or match$)	66	L60
L59	L58 and entity	73	L59
L58	L57 and relation	181	L58
L57	concept near information	497	L57
L56	L55 and relation	2	L56
L55	L53 and (numerical near value)	8	L55
L54	L53 and (graph near relation)	0	L54
L53	fuzzy near technique and graph	53	L53
DB=USPT; PLUR=YES; OP=OR			
L52	fuzzif$ near graph	0	L52
L51	L20 and membership	0	L51
L50	L30 and membership	3	L50
L49	L48 and parameter	6	L49
L48	L47 and (numerical near value)	11	L48
L47	l1 and relation	52	L47
L46	L45 and (numerical near value)	22	L46
L45	L44 and parameter	334	L45
L44	L43 and entitie	457	L44
L43	L42 and description	8997	L43
L42	audio near visual	9249	L42
L41	L40 and (numerical near value)	5	L41
L40	L39 and (parameter$)	51	L40
L39	L38 and (relation or relationship)	69	L39
L38	description near entity	95	L38
L37	L36 and (numerical near value)	1	L37
L36	L34 and (parameter$)	20	L36
L35	L34 and (parameter or parameters)	0	L35
L34	L33 and (relation or relationship)	37	L34

L33	L32 and ((entities or entity) near description)	39	L33
L32	((707/$)!.CCLS.)	11177	L32
L31	((707/$)!.CCLS.)	0	L31
L30	L27 and parameter	19	L30
L29	L27 and (description near parameter)	0	L29
L28	L27 and parameter	19	L28
L27	L26 and (numerical near value)	23	L27
L26	L23 and (description)	580	L26
L25	L24 and (numerical near value)	1	L25
L24	L23 and (description near information)	19	L24
L23	L21	580	L23
L22	L21	580	L22
L21	entities near relationship	580	L21
L20	L19 and graph$	16	L20
L19	L18 and parameter$	27	L19
L18	L17 and (numerical near value)	46	L18
L17	L15 and relation$	549	L17
L16	L15 and realtion$	0	L16
L15	L14 and description	847	L15
L14	((707/6)!.CCLS.)	860	L14
L13	L12 and (description near entities)	0	L13
L12	(updat$ near relationship)	187	L12
L11	l1 and (numerical near value)	11	L11
L10	L7 and (numerical near values)	8	L10
L9	L8 and (numerical near values)	7	L9
L8	L7 and relations	57	L8
L7	(description near entities)	157	L7
L6	L3 and (description near entities)	0	L6
L5	L3 and (description near schema)	0	L5
L4	L3 and relationship	28	L4
L3	L1 and description	52	L3
L2	L1 and (mpeg)	0	L2
L1	fuzzy near relation	52	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Thursday, August 21, 2003

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L65	L64 and link$	13	L65
L64	L63 and place and time	20	L64
L63	l32 and (wedding)	24	L63
L62	l32 and (wedding near video)	0	L62
L61	L60 and ((build$ or generat$) near graph)	4	L61
L60	L59 and (compar$ or match$)	66	L60
L59	L58 and entity	73	L59
L58	L57 and relation	181	L58
L57	concept near information	497	L57
L56	L55 and relation	2	L56
L55	L53 and (numerical near value)	8	L55
L54	L53 and (graph near relation)	0	L54
L53	fuzzy near technique and graph	53	L53
DB=USPT; PLUR=YES; OP=OR			
L52	fuzzif$ near graph	0	L52
L51	L20 and membership	0	L51
L50	L30 and membership	3	L50
L49	L48 and parameter	6	L49
L48	L47 and (numerical near value)	11	L48
L47	l1 and relation	52	L47
L46	L45 and (numerical near value)	22	L46
L45	L44 and parameter	334	L45
L44	L43 and entitie	457	L44
L43	L42 and description	8997	L43
L42	audio near visual	9249	L42
L41	L40 and (numerical near value)	5	L41
L40	L39 and (parameter$)	51	L40
L39	L38 and (relation or relationship)	69	L39
L38	description near entity	95	L38
L37	L36 and (numerical near value)	1	L37
L36	L34 and (parameter$)	20	L36
L35	L34 and (parameter or parameters)	0	L35
L34	L33 and (relation or relationship)	37	L34

L33	L32 and ((entities or entity) near description)	39	L33
L32	((707/$)!.CCLS.)	11177	L32
L31	((707/$)!.CCLS.)	0	L31
L30	L27 and parameter	19	L30
L29	L27 and (description near parameter)	0	L29
L28	L27 and parameter	19	L28
L27	L26 and (numerical near value)	23	L27
L26	L23 and (description)	580	L26
L25	L24 and (numerical near value)	1	L25
L24	L23 and (description near information)	19	L24
L23	L21	580	L23
L22	L21	580	L22
L21	entities near relationship	580	L21
L20	L19 and graph$	16	L20
L19	L18 and parameter$	27	L19
L18	L17 and (numerical near value)	46	L18
L17	L15 and relation$	549	L17
L16	L15 and realtion$	0	L16
L15	L14 and description	847	L15
L14	((707/6)!.CCLS.)	860	L14
L13	L12 and (description near entities)	0	L13
L12	(updat$ near relationship)	187	L12
L11	l1 and (numerical near value)	11	L11
L10	L7 and (numerical near values)	8	L10
L9	L8 and (numerical near values)	7	L9
L8	L7 and relations	57	L8
L7	(description near entities)	157	L7
L6	L3 and (description near entities)	0	L6
L5	L3 and (description near schema)	0	L5
L4	L3 and relationship	28	L4
L3	L1 and description	52	L3
L2	L1 and (mpeg)	0	L2
L1	fuzzy near relation	52	L1

END OF SEARCH HISTORY

WEST

Generate Collection Print

L12: Entry 1 of 11 File: USPT Aug 19, 2003

DOCUMENT-IDENTIFIER: US 6609118 B1
TITLE: Methods and systems for automated property valuation

Detailed Description Text (35):
The initial retrieval step 51 extracts a set of potential comparables using standard SQL queries for efficiency purpose. The selection of comparables is performed by comparing specific attributes of the subject property with the corresponding attribute of each comparable. All the comparables in the retrieved set have values within the allowable deviations. If the size of the retrieved set is too small (e.g., less than 10), the allowable deviations can be relaxed. For example, the initial retrieval can use the following attributes and their corresponding maximum allowable deviations (written after each attribute): Date of sale (within 12 months), Distance (within 1 mile), living area (+/-25%), lot size (+100%/-50%), Number of bedrooms (+/-3), Number of bathrooms (+/-3). These ranges correspond to the fuzzy sets 80 shown in FIG. 6 and the fuzzy relations 82, 84 shown in FIGS. 7 and 8. The first two attributes (number of months since the date of sale 86, and distance from subject 88) are market and region dependent, and each attributes range of allowed values are manually modified or automatically indexed to reflect low or fast markets, as well as urban, suburban, and rural regions. The remaining four attributes (living area 90, lot area 92, number of bedrooms(see FIG. 7), number of bathrooms (see FIG. 8)) reflect some of the subject property's main characteristics.

Detailed Description Text (36):
FIG. 6 describes the preference criteria for the first four attributes (number of months since the date of sale 86, distance from subject property 88, living area 90, and lot size 92). The trapezoidal membership distributions representing these attributes have a natural preference interpretation. For each attribute, the support of the distribution represents the range of tolerable values and corresponds to the interval-value used in the initial retrieval query. The core represents the most desirable range of values and establishes the top preference. By definition, an attribute value falling inside the core will receive a preference value of one. As the attribute value moves away from the most desirable range, its associated preference value will decrease from one to zero. At the end of this evaluation, each comparable has a preference vector, with each attribute having a preference value in the interval zero to one. These values represent the partial degree of membership of each attribute in the fuzzy sets and fuzzy relations for the preferred criteria. For example, by using the preference distributions shown in FIG. 6, the preference value for the attribute date-of-sale of a comparable that was sold within three months of today's date is one. If the date was six months ago, its preference value would be 2/3. Any comparable with a date of sale of more than twelve months would be given a preference value of zero. The remaining two features (number of bedrooms and number of bathrooms) are evaluated in a similar fashion. Their preference functions are represented by two reflexive asymmetric fuzzy relations, illustrated in FIGS. 7 and 8, respectively. For example, for a subject property with five bedrooms, the preferred comparable would also have five bedrooms (preference value=1), while a six bedroom comparable would meet that preference criterion to a degree of 0.8. Similarly, for a subject property with two bathrooms, the preferred comparable would also have two bathrooms (preference value=1), while a 2.5 bathroom comparable would meet that preference criterion to a degree of 0.7.

Detailed Description Text (43):
In addition to producing the final estimate of the value of the subject property, a reliability in the estimate is provided by attaching a reliability measure to each estimate. Ideally, the subject properties with the highest reliability will exhibit the lowest errors, and high reliability values are assigned to as many subject properties as possible. The reliability value is calculated from the following five quantitative

characteristics: number of cases found in the initial retrieval 262, average of the similarity values for the best four cases 264, typicality 266 of problem with respect to the case-base (i.e., if the attributes of the subject fall within typical ranges for the subjects five digit zip code region), span of, adjusted sales prices of highest reliability solutions 268 (i.e. the highest adjusted sales price minus the lowest adjusted sales price among the selected comparables), distribution of adjusted sales prices of highest reliability solutions 270 (i.e. average percentage deviation of the adjusted sales price of the comparables from the estimated value of the subject property). For example, these characteristics can be evaluated using the fuzzy membership functions 260 illustrated in FIG. 16. These functions map the numerical value of each parameter into a standard numerical reliability, which ranges from zero to one. These standardized reliability values are then aggregated into a final reliability value. Given the conjunctive nature of this aggregation, the minimum of the standardized reliability values is preferably used.